Cristea Constantin-Daniel -1405B

Zvoristeanu Alexandra-Ioana - 1405B

Tema:

Masina controlata de un server web.

Rezumat:

Proiectul nostru are ca scop realizarea unei masini ce poate fi controlata prin intermediul unei interfete web.
Prin apasarea unor butoane virtuale de catre utilizator, se ruleaza un script Python specific fiecarui buton, ce are ca efect trimiterea unui sir de caractere catre placuta XMC, iar aceasta, in functie de caracterele primite, controleaza miscarile masinii.

Resurse hardware:

- RaspberryPi model 3B
  https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
- Infineon XMC 1100- developement board
  https://www.infineon.com/dgdl/Infineon-xmc1100_AB-DS-v01_08-EN.pdf?fileId=5546d4624a0bf290014a4bdaff9325bd
- Motor de curent continuu

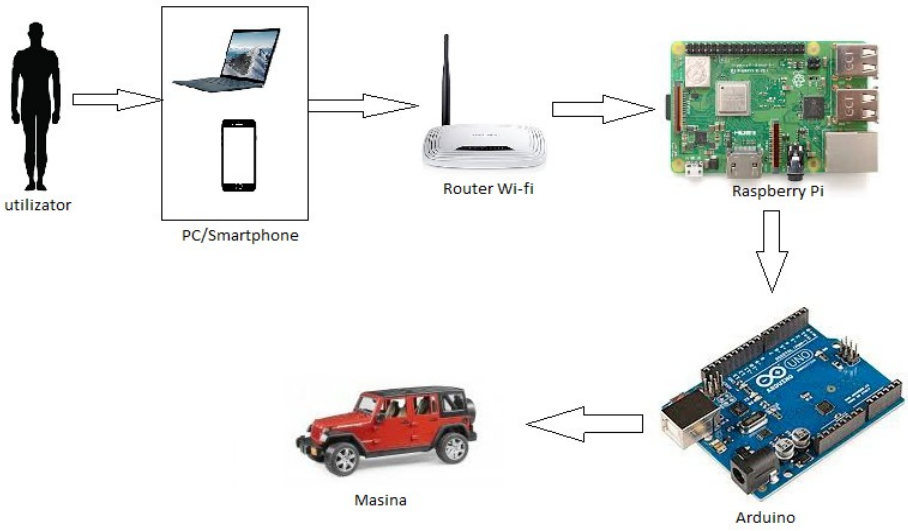
- Punte H
  https://hackerstore.nl/PDFs/Tutorial298.pdf
- Schelet masina
- Fire conectoare

Ciorna solutiei:

- RaspberryPi - developement board
- Infineon - developement board
- Motor de curent continuu
- Schelet masina
- Fire conectoare

Pentru a configura serverul Apache + PHP:
https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

Schema proiectului:



Cod dave:

```
#include <DAVE.h>

uint8_t data[] = "Infineon Technologies";
uint8_t rec_data[2];
```

```
char statu=1;
int main(void)
{
DAVE_STATUS_t status;

status = DAVE_Init();

if(status != DAVE_STATUS_SUCCESS)
{

XMC_DEBUG("DAVE APPs initialization faidir1\n");

}
else{

while(1){
statu=UART_Receive(&seriala,rec_data,sizeof(rec_data));

if(statu==UART_STATUS_SUCCESS)
{
//pentru servo-motor - directie
if(strcmp(rec_data,"A2")==0)
{
PWM_SetDutyCycle(&PWM_01, 920); //860
}
if(strcmp(rec_data,"A1")==0)
{
PWM_SetDutyCycle(&PWM_01, 800); //800
}
if(strcmp(rec_data,"A0")==0)
{
PWM_SetDutyCycle(&PWM_01, 680);//740
}


//sens de mers + stop
if(strcmp(rec_data,"B0")==0)
{
```

```
DIGITAL_IO_SetOutputHigh(&dir1);
DIGITAL_IO_SetOutputLow(&dir2);
DIGITAL_IO_SetOutputHigh(&stop);

}
if(strcmp(rec_data,"B1")==0)
{
DIGITAL_IO_SetOutputLow(&dir1);
DIGITAL_IO_SetOutputLow(&dir2);
DIGITAL_IO_SetOutputLow(&stop);
}
if(strcmp(rec_data,"B2")==0)
{
DIGITAL_IO_SetOutputLow(&dir1);
DIGITAL_IO_SetOutputHigh(&dir2);
DIGITAL_IO_SetOutputHigh(&stop);

}

//faruri
if(strcmp(rec_data,"C0")==0)
{
DIGITAL_IO_SetOutputHigh(&far);

}
if(strcmp(rec_data,"C1")==0)
{
DIGITAL_IO_SetOutputLow(&far);
}

//sirena
if(strcmp(rec_data,"D0")==0)
{

int k =0;
while(k<10){
BUS_IO_Write(&BUS_IO_0,0x0F);
BUS_IO_Write(&BUS_IO_1,0xF0);
```

```
for(int i=0;i<0xaffff;i++);
BUS_IO_Write(&BUS_IO_0,0xF0);
BUS_IO_Write(&BUS_IO_1,0x0F);
for(int i=0;i<0xaffff;i++);
k++;
}
BUS_IO_Write(&BUS_IO_0,0x00);
BUS_IO_Write(&BUS_IO_1,0x00);

}
if(strcmp(rec_data,"D1")==0)
{
BUS_IO_Write(&BUS_IO_0,0xFF);
BUS_IO_Write(&BUS_IO_1,0xFF);
}
}
}
}
}
```